



Leveraging Antenna Orientation to Optimize Network Performance of Fleets of UAVs

Rémy Grünblatt, Isabelle Guérin-Lassous, Olivier Simonin

► To cite this version:

Rémy Grünblatt, Isabelle Guérin-Lassous, Olivier Simonin. Leveraging Antenna Orientation to Optimize Network Performance of Fleets of UAVs. MSWiM'20 - The 23rd International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Nov 2020, Alicante, Spain. 10.1145/3416010.3423225 . hal-02934386

HAL Id: hal-02934386

<https://inria.hal.science/hal-02934386>

Submitted on 3 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Leveraging Antenna Orientation to Optimize Network Performance of Fleets of UAVs

Rémy Grünblatt
remy@grunblatt.org
Univ Lyon, EnsL, UCBL, CNRS, Inria,
LIP & CITI, France

Isabelle Guérin Lassous
isabelle.guerin-lassous@ens-lyon.fr
Univ Lyon, EnsL, UCBL, CNRS, Inria,
LIP, France

Olivier Simonin
olivier.simonin@insa-lyon.fr
Univ Lyon, INSA Lyon, Inria,
CITI, France

ABSTRACT

In this paper, we investigate the problem of optimizing the network performance of a fleet of unmanned aerial vehicles (UAVs) in static positions. More precisely, we allow each UAV to change its orientation in order to improve the quality of communication with its neighbours. This form of controlled mobility takes advantage of the effective radiation pattern of each UAV. We build a decentralized scheme based on the hill climbing optimization approach without *a priori* knowledge of the antennas radiation patterns. Then, we propose a simulation framework, based on ns-3, allowing to evaluate the gain in network performance. We provide results in several deployment scenarios involving different rate adaptation algorithms and network sizes.¹

ACM Reference Format:

Rémy Grünblatt, Isabelle Guérin Lassous, and Olivier Simonin. 2020. Leveraging Antenna Orientation to Optimize Network Performance of Fleets of UAVs. In *23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '20)*, November 16–20, 2020, Alicante, Spain. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3416010.3423225>

1 INTRODUCTION

Fleets of unmanned aerial vehicles (UAVs) can be used for search and rescue missions, emergency deployment of communication networks, surveillance, or even natural phenomena monitoring. While these different applications have different needs in terms of communication quality, throughput, coverage or infrastructure, the IEEE 802.11 set of standards is a good candidate to enable communication throughout the fleet and between the fleet and the ground. Indeed, WiFi supports both ad-hoc and infrastructure modes, is readily available and can be used without license worldwide, while being the de-facto standard wireless communication technology for a whole set of devices such as laptops or phones. Moreover, the latest WiFi version (WiFi5) (or the upcoming version WiFi6) offers (or will offer) very high throughputs enabling to consider the transmission of high demanding flows within UAV networks.

Experiments regarding UAV 802.11 networks and their air-to-ground and air-to-air links underline that antenna radiation patterns play a big role in the performance of the deployed network

¹The authors would like to thanks the Direction Générale de l'Armement (DGA) and the Fédération Informatique de Lyon (FIL) for their financial support.

Publication rights licensed to ACM. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of a national government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.

MSWiM '20, November 16–20, 2020, Alicante, Spain

© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-8117-8/20/11...\$15.00

<https://doi.org/10.1145/3416010.3423225>

[9, 16, 17]. This conclusion is also highlighted in a very recent work [1] that, via experiments, shows the impact of different physical parameters on the UAV communication performance. Yet, theoretical works and simulations often assume that omnidirectional antennas (also called isotropic antennas) are used in UAV networks. Whether the use of directional antennas can improve the communication performance within UAV networks and how to take into account these directional antennas are still, from our point of view, open questions. In this paper, we explore how to leverage the orientation of directional antennas in order to improve the network performance of a fleet of UAVs. The main contributions of this paper are summarized as follows:

- We propose a controlled mobility algorithm based solely on the orientation of the UAVs, without an *a priori* knowledge of the radiation pattern of the used antennas. The proposed algorithm is distributed and each UAV asynchronously runs its own algorithm based on local measurements on the power of the received packets.
- We have developed and implemented a simulation framework to simulate a network of UAVs equipped with directional antennas of any possible radiation pattern. The simulation framework consists of two main components: one component simulates the UAVs and the orientation of their antenna while the second component is based on the ns-3 network simulator to simulate the network communications. The interactions between the two components are realized via a message-passing approach.
- Based on this simulation framework, we have evaluated the proposed antenna orientation solution. Different network topologies with different number of UAVs are considered with several rate adaptation algorithms. The solution convergence time, the obtained throughput for each transmitted flow as the network Jain's index have been studied.

The paper is organized as follows: the studied problem is formally modeled in Section 2. From this modeling, we describe the proposed antenna orientation algorithm in the same section. Then, the simulation framework developed to simulate UAV and antenna orientation is described in Section 3. In Section 4, we describe the different scenarios that have been evaluated as the obtained results. Papers that relate to the studied problem are discussed in Section 5. We conclude in Section 6.

2 PROBLEM MODELING AND OPTIMIZATION OF ANTENNA ORIENTATION

In this section, we first introduce the studied problem and we then describe the proposed solution for the antenna orientation.

We consider a set of UAVs (also named as agents or nodes hereafter), each equipped with a wireless network interface controller using WiFi and a directional antenna whose the radiation pattern (also named the antenna gain pattern) is unknown. All the agents use the same Wi-Fi channel to communicate. The studied problem is the following: given a UAV fleet configuration, can each agent optimize its local antenna orientation in order to enhance the communication performance, such as throughput? We focus on multi-rotor UAVs because their three-dimensional positions and orientations can be fully controlled and maintained through time by the flight controller, while, for example, fixed-wing UAVs cannot hover at a given position. We also limit this study to UAVs whose 3D positions are static, but whose orientations in their horizontal plane, around the normal axis, named yaw, can be changed. Indeed, as the 3D UAVs positions are often application dependent, we focus on parameters that can be modified without interactions with the applications, for the sake of generality. These requirements cover, in particular, the class of coverage applications, such as surveillance, continuous monitoring or network coverage. As changing the roll (orientation along the longitudinal axis) or the pitch (orientation along the transverse axis) of a multi-rotor changes its 3D position when it is not subject to external forces apart from gravity, we assume those two quantities are also fixed.

In this study, we want to optimize the overall network throughput by changing the agent antenna orientations when the agents are in fixed positions. The throughput obtained by each transmitted flow in the network depends, among others, on the transmission rate used to transmit the flow, the quality of the channel used for the communication, the transmission power and the antenna types and orientations. The transmission rate is very often adaptive and regulated by a rate adaptation algorithm (also noted RAA hereafter). There exist many different rate adaptation algorithms [14] and a large number of Wi-Fi interfaces use proprietary solutions for which the used algorithm is unknown. These algorithms can lead to very different performance for the same scenario, as shown in [7]. We thus think that it is difficult to design a generic antenna orientation solution leveraging the used rate adaptation algorithm. We have therefore decided to use, in our solution, a simpler metric like the power of the received signal. This metric is impacted by the antenna orientation and the channel quality, but it also has an indirect impact on the used rate adaptation algorithm. In Section 4, we will study the performance of our proposed solution with different rate adaptation algorithms.

2.1 Problem Modeling

Let $G = \{V, E\}$ be an undirected graph representing a set of N networked agents, where $V = \{A_1, A_2, \dots, A_N\}$ and $E \subset V \times V$ denote respectively the set of vertices and the set of edges. We denote by $Ad = (a_{i,j})_{(i,j) \in N \times N}$ the adjacency matrix of the graph: $a_{i,j} = 1$ if $(i, j) \in E$ meaning that agents A_i wants to communicate with agent A_j , and $a_{i,j} = 0$ if $(i, j) \notin E$ meaning that agent A_i does not wish to communicate with agent A_j .

Each agent is equipped with a directional antenna. The antenna radiation pattern is represented by a function g . As g can be different from one agent to another, we use g_i representing the antenna radiation pattern of agent A_i . It is expressed in decibels and in the

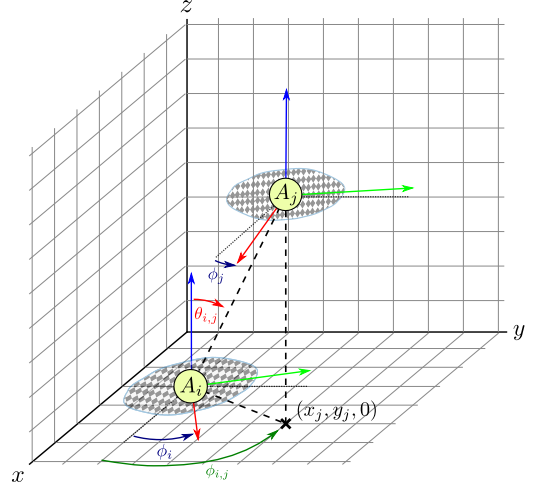


Figure 1: 3D view of two agents A_i and A_j

spherical coordinates system described in [3, Chapter 2.2]. Figure 2 gives an example of two antenna gain patterns in a plane (one directional antenna and one omnidirectional antenna). Depending on the antenna orientation between two neighbor agents (there exists a link between these 2 agents in G), these two agents may be able to communicate or not. When they are able to communicate, this orientation has also an impact on the power of the received signal. The higher the received power, the more likely the communication will be of good quality and will use a high transmission rate.

The objective of the agents is then to cooperatively solve the following optimization problem:

$$\max_{\phi \in [0, 360]^N} f(\phi) := \sum_{i \in \{1, \dots, N\}} \sum_{\substack{j \in \{1, \dots, N\} \\ j \neq i}} a_{i,j} * S_{i,j}$$

with

$$S_{i,j} = e_j + g_j(\pi - \theta_{i,j}, \phi_{i,j} + \pi - \phi_j) + g_i(\theta_{i,j}, \phi_{i,j} - \phi_i) - C_{i,j}$$

if $S_{i,j} \geq Th$

$$= 0 \text{ otherwise}$$

$S_{i,j}$ represents the received power, at agent A_i , of the signal sent by agent A_j and ϕ is the yaw orientation vector giving the yaw orientation of each agent (ϕ_i is the yaw orientation of agent A_i). The scalar e_j represents the transmission power of agent A_j in dBm and the scalar $C_{i,j}$ represents the loss induced by the channel between agents A_j and A_i , in dB. The antenna gains used during the communication between agent A_i and agent A_j depend on their position and their relative orientation.

Assuming agent A_i is located at (x_i, y_i, z_i) and agent A_j is located at (x_j, y_j, z_j) , we have

$$\theta_{i,j} = \text{atan2}(\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}, z_j - z_i)$$

and

$$\phi_{i,j} = \text{atan2}(y_j - y_i, x_j - x_i)$$

which represent respectively the relative polar and the relative azimuth angles between agents A_i and A_j . These quantities are represented on Figure 1. $S_{i,j}$ is a non null value if $S_{i,j}$ is higher than

a given threshold Th representing the minimal signal-to-noise ratio required to receive data.

Finding a solution to this optimization problem involves determining the different agent antenna orientations to optimize the sum of the powers of the received signals in the network. In the next section, we propose a distributed solution in which each agent determines its antenna orientation based on local measurements, without knowing its antenna gain pattern nor the ones of the other agents, or their positions.

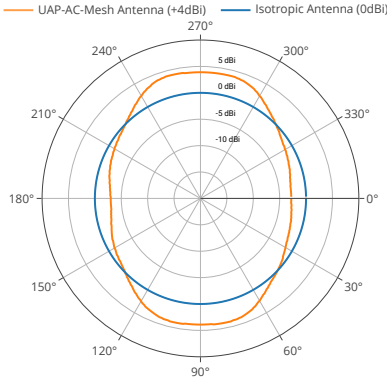


Figure 2: Radiation pattern of the antennas used during the simulations for $\theta = 90^\circ$ (horizontal plane), in dBi (decibel relative to the isotropic antenna).

2.2 Optimization of Antenna Orientation

As the explicit expression of g is unknown from the agents, the proposed solution will be based on measurements that each agent can carry out. More precisely, agent A_i can measure $S_{i,j}(t)$ at time t if the following conditions are met: agent A_j is transmitting at time t , $a_{i,j} = 1$ and $S_{i,j}(t)$ is bigger than the given threshold Th (for the SNR). When agent A_i carries out such a measurement, it knows its yaw orientation $\phi_i(t)$. These measurements will be stored in a measurement vector M : each agent A_i maintains $M_{i,j} = [m_{i,j,0}, m_{i,j,1}, \dots, m_{i,j,359}]$ for each agent A_j such that $(i, j) \in E$. The scalar components $m_{i,j,k}$ corresponds to the measurement of the mean received power, at agent A_i , of the signals sent by agent A_j when A_i has a yaw orientation equals to k . Because we are not requiring the knowledge of G and E from the agent A_i , $M_{i,j}$ is created "on the fly" when the connection between A_i and A_j is first established. It is then initialized to $M_{i,j} := [\text{NONE}, \dots, \text{NONE}]$.

Each agent executes its own algorithm without being synchronized with its neighbors. The proposed algorithm consists of an infinite loop. In each passage in the loop, each agent realizes different steps. First, the agent fetches the frames it has received since the last loop execution, in its network interface queue, and updates its measurement vectors. Then, if the agent lacks some data in its measurement vector with at least one neighbor, it seeks which orientation to move to, to get this measurement. Finally, if its measurement vectors are complete, it tries to optimize its orientation based on their values. Each agent runs the algorithm while it changes its orientation according to online results and while it communicates with its neighbors if required by the data traffic.

The proposed algorithm is based on the hill climbing approach [15]. We have chosen hill climbing for two reasons: 1) it is an anytime algorithm (it can find better and better solutions as long as it keeps running) and 2) even if it does not guarantee convergence towards a global optimum, it provides an efficient way to find a good solution in a decentralized multi-agent problem. Algorithm 1 describes the algorithm executed by agent A_i .

Algorithm 1: ANTENNA ORIENTATION OPTIMIZATION (agent A_i)

```

1  $Window \leftarrow 360$  ;  $Goal \leftarrow \text{None}$  ;  $Count \leftarrow 0$  ;
2 while true do
    % Measurement updates
3 if  $m_{i,j,\lfloor \phi_i(t) \rfloor} = \text{NONE}$  then  $m_{i,j,\lfloor \phi_i(t) \rfloor} \leftarrow S_{i,j}(t)$ 
4 else  $m_{i,j,\lfloor \phi_i(t) \rfloor} \leftarrow \text{mean}(m_{i,j,\lfloor \phi_i(t) \rfloor}; S_{i,j}(t))$ 
    % New orientations to explore
5 if there exists  $k$  such that there exists  $j$  such that  $m_{i,j,k} = \text{NONE}$ 
6 then
7     find the  $k$  minimizing  $|\lfloor \phi_i(t) \rfloor - k|$ 
8      $Goal \leftarrow k$ 
9     if  $k == \phi_i(t)$  then  $Count \leftarrow Count + 1$ 
10    else  $Count \leftarrow 0$ 
11    if  $Count \geq 10$  then
12         $Count \leftarrow 0$ 
13         $m_{i,j,k} \leftarrow -100$  for any  $j$  such that  $m_{i,j,k} == \text{NONE}$ 
    % Orientation optimization
14 if there exists no  $j$  or  $k$  such that  $m_{i,j,k} = \text{NONE}$  and ( $Goal$  is
15  $\text{NONE}$  or  $Goal == \lfloor \phi_i(t) \rfloor$ ) then
16     Find  $l$  in  $[\lfloor \phi_i(t) \rfloor - Window/2; \lfloor \phi_i(t) \rfloor + Window/2]$ 
        maximizing:
        
$$\sum_{j \in [1:N]} a_{i,j} * m_{i,j,l}$$

17      $Goal \leftarrow l$ 
18      $Window \leftarrow Window/2$ 
    % Change of orientation
19 Set  $d_i$  to reach  $Goal$ 

```

As the orientation $\phi_i(t)$ and the power measurement $S_{i,j}(t)$ of the received signal depend on the instant at which these 2 parameters are considered, they are expressed in function of the time t . The *Window* variable represents the search space. Initially, the search space includes all the possible orientations ($[0; 360[$). In order to speed up the algorithm convergence, the size of the space search is divided by 2 as soon as a maximal solution is found in the current space search (line 18 of Algorithm 1). The *Count* variable represents the maximum number of loop passages during which the agent stays in the same orientation. If the agent stays in a given orientation for too long while trying to fill its measurement vector, the agent considers that it is not a good orientation and sets a very low value to the corresponding measurement element (line 13 of Algorithm 1). The *Goal* variable represents the orientation the agent is currently trying to reach. In the first loop passages, *Goal* corresponds to unexplored orientations for which no measurement

value has been collected. Once measurements have been collected for all the orientations and neighbors, then an optimal orientation (in respect to the defined objective function) can be computed. Then, the parameter d_i , representing the direction to follow (*i.e.* right, left, or do not move), is updated in order to reach the orientation *Goal*.

Note that finding an optimal orientation does not mean the end of the algorithm. The search continues with new possible measurements and on a reduced search space. A new optimal solution can thus be found.

Evaluating this algorithm is a difficult task because the algorithm is distributed and executed in parallel by all the agents in an asynchronous way, but also because it depends on the data traffic, the medium access, the used transmission rates, the channel quality and the agent controller. Moreover, we are interested in the network performance. A dedicated simulation framework has therefore been developed to evaluate the proposed antenna orientation solution in a realistic context.

3 SIMULATION FRAMEWORK

To simulate antennas and UAVs and to evaluate our proposition, we chose to develop a framework based on the ns-3 network simulator. This development is necessary as existing simulation frameworks do not offer the possibility to both simulate UAV controllers and the UAV communications using directional antennas easily. While ns-3 has some support for antenna modelling, this support is only compatible with Long-Term Evolution networks and not with WiFi networks. Nodes, which represent the physical objects in ns-3, are point-like objects with Cartesian coordinates, but no orientation coordinates are provided. These two facts together make it difficult to simulate spinning nodes embedding non-isotropic antennas for WiFi networks, at least without rewriting much of the ns-3 models. UAV simulators based on ns-3 such as CUSCUS [18] or FlyNetSim [2] are focused on hardware-in-the-loop, software-in-the-loop or real-time simulations, and do not model the antennas. The developed simulation framework is available as an open source project at GitHub².

3.1 Architecture

The architecture of our framework, as depicted in Figure 3, is divided into two main components. The first component is the network simulator ns-3, including the user simulation script or program (bottom left), backed by a custom ns-3 module implementing a *propagation loss model* and a *mobility model* (bottom right), and the second component is the discrete-event antenna and UAV simulator, called PHI (top). A third optional component, the visualization front-end, can be plugged into the PHI simulator in order to follow the state of the simulations in "real" (simulated) time.

The goal of PHI is to simulate the behaviour and dynamics of multiple UAVs equipped with non-isotropic antennas. PHI provides, according to the antenna orientation, the power gains to use in the ns-3 simulator, simulator that will in turn simulate the UAV network and the networking stack. The controllers and the sensors of the UAVs are therefore modelled by PHI. The simulator has been implemented in C++, which is also the language used by ns-3, but

as the interface between PHI and ns-3 relies on message passing, the language could easily be changed.

Communications. The two components (ns-3 and PHI) exchange Protocol Buffers messages, used to serialize and deserialize structures, and communicate using the Zero MQ asynchronous messaging library. Interactions between ns-3 and PHI take the form of two types of messages: META messages that are used to set up and control the simulations life cycle, and MESO messages that concern the simulation itself. The socket connecting the two components uses the ZeroMQ request-reply pattern, ensuring their synchronization, and currently uses a local inter-process communication transport. A third type of message, called Viz, serves as a way to serialize the state of the simulation in order to send it to the visualization front-end.

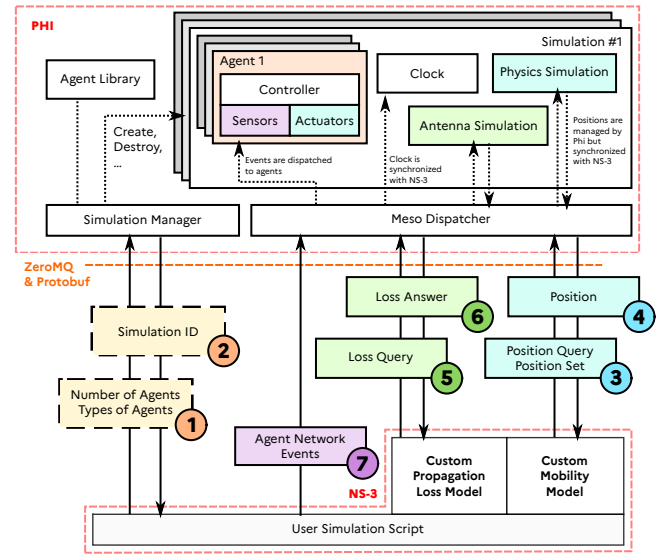


Figure 3: High level overview of the simulation framework architecture: main components and control and data paths.

Simulation Life Cycle. When a simulation is set up in ns-3, a META message ① containing the number of agents and their types are sent to PHI's simulation manager component, which will instantiate the simulation, and reply with a simulation ID ②. This simulation ID is included in every subsequent META or MESO messages exchanged between the two components and allows a single instance of PHI to be used by concurrent ns-3 simulations. When the simulation ends in ns-3, a META message is sent to the PHI's simulation manager to end the simulation and release the resources.

Mobility Model and Physics Engine. In ns-3, *mobility models* are in charge of tracking and changing nodes' positions, speeds and accelerations. These quantities are used by the propagation loss and delay models to compute a loss depending on the distance between nodes, and to compute the delay between the transmission and reception of the frames. A custom ns-3 mobility model has been developed, allowing to set the position in PHI from the ns-3 simulation ③, for example at the beginning of a scenario, as well as

²Phi codebase: <https://github.com/rgrunbla/Phi>.

querying node positions ④, e.g. to calculate propagation delays. As the simulator was developed for scenarios where the UAVs have a static position and dynamic orientation, we chose to only consider constant rotation speeds for the UAVs. This allows to model their movements and rotations with simple multiplication operations without having to go through the use of a differential equation solver. This approach is also used by ns-3, which only supports constant speed or constant acceleration mobility models.

Propagation Loss and Delay Model. In ns-3, *propagation loss models* and *propagation delay models* are used to model the propagation of the signal between any two nodes, by respectively calculating the signal power and the signal delay. These models can be chained, for example adding a model of Nakagami fading to a free space path loss model leading to a link budget calculation performed by the channel model. The custom module implements a propagation loss model which queries PH1 ⑤ about the gains brought by the antennas of the agents, gains which are sent back to ns-3 ⑥. PH1 does not model any other effect such as the free space loss as one can use the ns-3 models directly. No custom propagation delay model is needed, as this calculation can be done by ns-3 directly by using the positions set by the custom *mobility model*.

Clock Synchronization. The clock state of a ns-3 simulation is included into every MESO message sent to PH1, for example in a propagation loss query, or a position query. The only way for a simulation in PH1 to advance through time is to receive a MESO message and synchronize its clock with the value it contains. Before the clock update, all the events in the event queue of PH1 that are scheduled to occur before the new clock value are executed, with each event being preceded by an update of the environment and agents states.

Agent Simulation and Environment. Each agent simulated by PH1 is specified by a type and the associated blueprint located in the *Agent Library* component. This blueprint contains the implementation of the controller, of the sensors and the actuators. These components are functions executed at their own frequency using events, e.g. 100 Hz for the controller or 10 Hz for a magnetometer. The controller is only capable of interacting with its environment through the use of a shared memory with sensors and actuators, in an asynchronous way. Messages originating from the ns-3 simulation intended for a specific agent are called *Network Events* ⑦ and are placed in a queue in the shared memory. Such messages are for example sent by ns-3 when a frame is received, and contain the frame characteristics, such as the reception power, or the MAC address of the transmitter if applicable.

4 EVALUATION

In this section, we present different scenarios used to evaluate the performance of our approach (Algorithm 1). The different scenarios share some parameters, described in Table 1, but differ in the number of nodes and their positions. We use the ns-3 Friis propagation loss model, also known as the free-space path loss model, which accurately models the path loss of air-to-air communications between UAVs [17]. All of the simulations rely either on an isotropic antenna or a directional antenna whose orientation is regulated by Algorithm 1. The directional antenna represents the Ubiquiti

Simulation Parameter	Value
Simulation Duration	100s
Wifi Standard	802.11 ac
Wifi MAC type	Ad-Hoc
Rate adaptation algorithm	MinstrelHt, Ideal or Intel
Spatial Streams	2x2:2
Channel Width	20 Mhz
Antennas	Isotropic or Ubiquiti UAP-AC-Mesh
Propagation Loss Model	Friis
Routing	Static
Application	Constant bitrate, UDP
UAV Rotation Speed	0 rad/s, ± 0.50 rad/s
Controller Frequency	100 Hz
Magnetometer Frequency	10 Hz

Table 1: ns-3 simulation parameters

UAP-AC-Mesh Antenna, also named as mesh antenna hereafter, whose radiation pattern is shown on Figure 2. The radiation pattern is provided by the constructor on its website [13] as an *ant* file type, covering the horizontal plane with a granularity of 1° . This antenna has been chosen for its small size and weight, making it compatible with airborne applications, as well as its balanced radiation pattern suitable for mesh applications. The tested antenna has a maximal gain of 4 dBi. Considering a link with two agents equipped with the directional antenna with a 4 dBi gain, 63% of all the possible orientations between the two agents yield a higher gain than a link with two isotropic antennas.

Three rate adaptation, named MinstrelHT, Intel and Ideal, have been considered. MinstrelHT is implemented in the mac80211 component of the Linux kernel and is open source [5]. MinstrelHT is used in the ath9k driver. The Intel rate adaptation algorithm is used on Intel WiFi interfaces and on Intel Aero Ready-to-Fly UAV [7]. Ideal is another rate adaptation algorithm implemented in ns-3 and supporting 802.11ac. These three algorithms have different behaviors and lead to different performance, as studied in [7]. Our evaluation will thus also study the impact of these three algorithms on the performance of our solution.

We present the obtained results in Sections 4.1, 4.2 and 4.3. The initial orientations of the nodes are distributed uniformly over $[0; 2\pi]$, and each simulation is repeated 20 times with different initial orientations. Several results are reported with the box plot representation.

4.1 Scenario #1: Simple

In this scenario, two nodes are separated by a fixed distance, with one node acting as a source and one node acting as a sink, as shown on top of Figure 4. The two nodes are either both equipped with omnidirectional antennas, in which case Algorithm 1 is not used, or both equipped with directional antennas using our antenna orientation algorithm. The throughput of the source is set to 180 Mbps, which exceeds the maximum physical throughput for the WiFi physical layer parameters used in the simulation, which is 173.3 Mbps. The received throughput at the sink is plotted on Figure 5 as a function of time, rate adaptation algorithm, and the used antenna, for a single simulation and when the distance between the

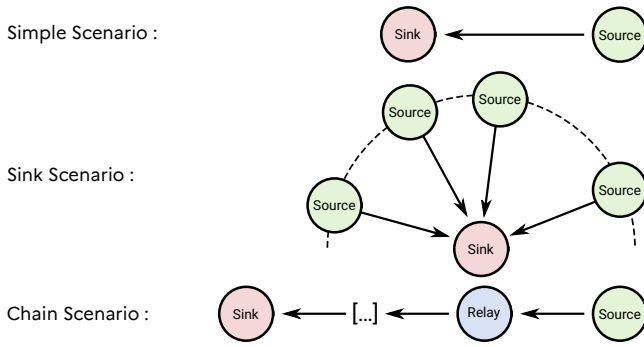


Figure 4: Overview of the studied scenarios

two nodes is 100m. We can observe that the three rate adaptation algorithms are almost the same when an omnidirectional antenna is used, and the received throughput remains stable throughout the simulation (with some variations with MinstrelHT). When the directional antenna is used, we observe two main phases. The first phase, where the throughput varies a lot corresponds to the execution of the antenna orientation algorithm: as the channel between the two nodes changes, the rate adaptation algorithms react and change the transmission rates, affecting the received throughput. The second phase starts after the antenna orientation algorithm has converged to its best solution in terms of received power. The received throughput remains fairly stable during this phase as the only source of change is the RAA decisions. We can however observe that when MinstrelHt is used, it takes more time to reach the stabilized received throughput, which is consistent with the previous observations made about the algorithm in [7].

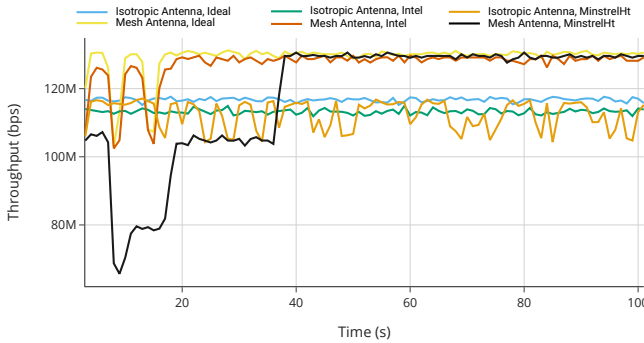


Figure 5: Evolution of the application throughput in function of time for Scenario #1, with 2 nodes 100m apart and a saturating UDP application rate of 180 Mbps.

The convergence time for the antenna orientation algorithm and the convergence time on the received throughput for the simulations using the directional antenna are plotted on Figure 6 (with the box plot representation). The convergence time for the antenna orientation algorithm is the elapsed time between the start of the algorithm and the time when the last agent stops to change its orientation. The convergence time on the received throughput is the elapsed time between the start of the algorithm and the time when the received throughput on the sink is different to at most

5% of the final achieved received throughput. We can note that the convergence time of our algorithm is always smaller than 20s in Scenario #1. The convergence time on the received throughput is also smaller than 20s for Ideal and Intel, and it is never larger than 40s with MinstrelHT. We can observe a strong correlation between the two quantities for the Ideal and the Intel RAAs, which underlines those algorithms are fast to react to changes in the channel, while the throughput convergence time of MinstrelHt illustrates the inertia of the algorithm, which can be linked to its sampling approach. Using physical layers metrics, such as signal strength, and not application layer metrics, such as the received throughput, appears therefore justified for such an algorithm, as higher layer metrics may introduce important delay with certain RAA.

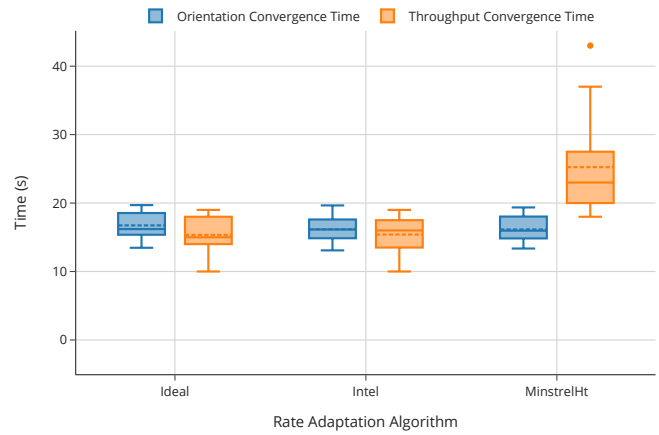


Figure 6: Comparison of the convergence time for the antenna orientation algorithm and application throughput for Scenario #1 at $d = 100m$.

Figure 7 shows the distribution of the achieved throughput for Scenario #1 when the two nodes are 100m away. The achieved throughput is measured when the antenna orientation has converged. The obtained results show that our antenna orientation solution improves the achieved throughput whatever the used RAA. For instance, with the Ideal RAA, the mean achieved throughput is 144.2 Mbps with directional antennas compared to 116.8 Mbps with omnidirectional antennas, whereas it is 136.9 Mbps with directional antennas compared to 113.2 Mbps with isotropic antennas for the Intel RAA. For MinstrelHT the use of directional antenna with our orientation algorithm leads to 137.4 Mbps compared to 111.5 Mbps with omnidirectional antennas. We analyzed the antenna orientations obtained when our algorithm has converged for the different simulation repetitions and for the different RAAs. The values obtained on the antenna orientations vary but are mainly scattered on good positions as 95% of the achieved orientations lead to a better link budget than with isotropic antennas. These orientations lead to better link qualities which also lead to a use of higher transmission rates, which, at the end, results in higher achieved throughputs. Finally, one can note that the obtained values on the throughput are more dispersed with directional antennas than with isotropic antennas. This is explained by the fact that the obtained orientations vary, which results in different link budgets

implying more various throughputs, though these latter are, most of the time, better than the ones obtained with isotropic antennas.

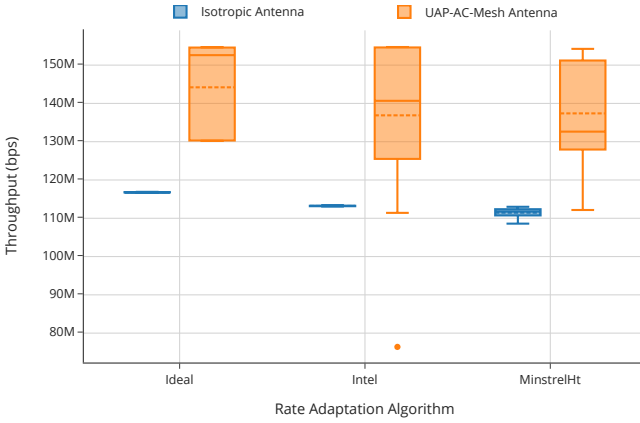


Figure 7: Comparison of the achieved throughput for Scenario #1 at $d = 100m$.

4.2 Scenario #2: Sink

In this scenario, one node serves as a sink while other nodes serve as sources. The sources are located on a circle with a fixed radius r , while the sink is located at the center of the circle, as shown on the middle of Figure 4. The sink can be seen as a UAV receiving video feeds from the sources, and sending them to the ground using another network component not studied here. The sink is equipped with an isotropic antenna. We have observed, on the different simulations, that the antenna orientation algorithm converges in less than 30s. The distribution of the average received throughput per link, at the sink, for a radius of 100m and for different application rates at the source, is shown on Figure 8. One can observe an increase in the obtained throughput when using the directional antenna, no matter which RAA is used. The increase is more limited with MinstrelHT. As 100% of the simulations obtain a better link budget than with an omnidirectional antenna, this more limited improvement can be explained by the larger time needed for MinstrelHT to converge towards the final throughput when the antenna orientation has ended, leading to a smaller throughput than with Ideal and Intel. When the application rate is low enough, e.g. 10 Mbps, it can be fulfilled by both the isotropic and the directional antennas in any direction, leading to very similar obtained throughput.

We have measured whether the different sources are receiving a fair "share" of the received throughput at the sink or not with the Jain's fairness index [10]. The results obtained for $r = 100m$ and $n = 3, 5, 10$, and an application throughput of 50 Mbps show that the use of the mesh antenna does not decrease the fairness between the nodes with rather a slight increase of the Jain's index values (between 0.94 to 1 for directional antennas compared to 0.92 to 1 with isotropic antennas).

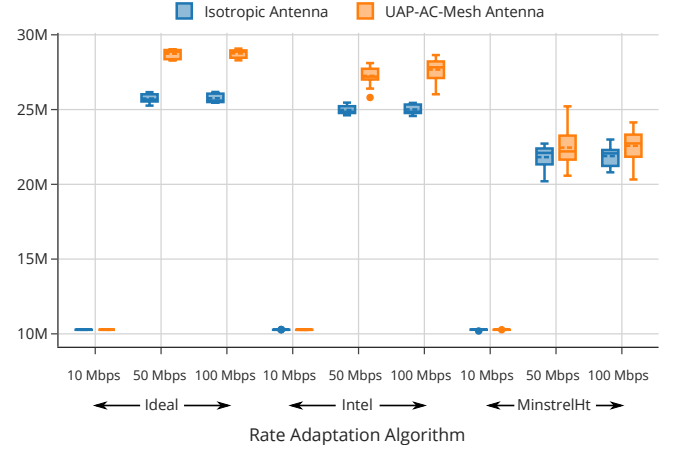


Figure 8: Comparison of the average received throughput per link for Scenario #2 with $r = 100m$ and 5 nodes.

4.3 Scenario #3: Chain

In this scenario, one node serves as a source, one node serves as a sink, and the other nodes serve as relays between the source and the sink as depicted in the bottom of Figure 4. The source and the sink are separated by a fixed distance d , and the relays are equidistantly placed between them.

We plot the distribution of the received throughput at the sink on Figure 9, for a distance between the source and the sink of $d = 1000m$, and for 5 and 10 nodes in total, that is to say for respectively 3 and 8 relays, for an application throughput of 50 Mbps. While for a chain of 5 nodes, the use of the directional antenna with the antenna orientation algorithm improves the overall throughput, for any RAA, no improvement is observed for a chain of 10 nodes with the Intel and MinstrelHT RAAs.

We observe that while the percentage of failed MAC transmissions at the source is higher with the directional antenna, the frame transmission rate also increases, leading to lower air-time per frame, allowing more frames to be exchanged. This property is also verified on the different links of the chain. This results in higher throughput with directional antennas than with omnidirectional ones. On the other hand, with 10 nodes, a too high number of retransmissions has been observed whatever the used antenna, leading to low throughput in both cases. One can also note that Intel exhibits poor performance, in this scenario, compared to Ideal and MinstrelHT. It can be explained by the conservative behavior of the Intel RAA when too many retransmissions are triggered [7].

5 RELATED WORK

The interest of using directional antennas in UAV networks has been shown in some experiments. In [8], the authors show that, when using Wi-Fi directional antennas, 2 UAVs can communicate with an acceptable throughput (of the order of several Mb/s) even if the distance between the 2 UAVs is large (around 1 km). In these experiments, IEEE 802.11g is used and there is no indication on the used rate adaptation algorithm or if any such algorithm is disabled. In [1], the authors experimentally show the impact of the antenna orientations on the UAV communications. Different effects are studied like the UAV body, the UAV relative direction and the

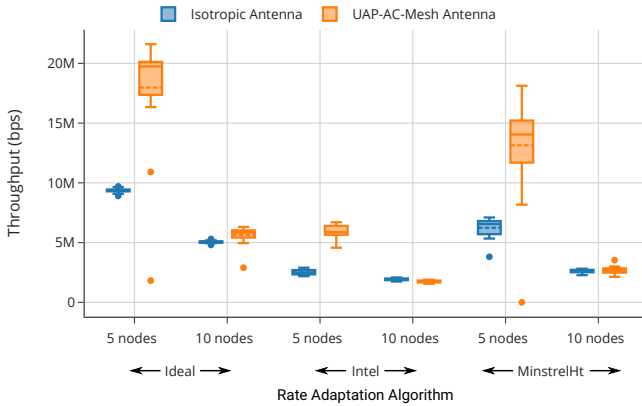


Figure 9: Comparison of the average received throughput at the sink for Scenario #3 with $d = 1000\text{m}$, 5 and 10 nodes.

elevation angle on different parameters like the Received Signal Strength and the cross-polarization discrimination. The obtained results show the importance of the antenna orientation and their impact on the expected communication performance.

In [4], an antenna heading control system is proposed for UAVs equipped with directional antenna. This system is mainly based on GPS information, but also on a RSSI (Received Signal Strength Indicator) scan when GPS data are not available. The directional antennas are used for the communications between UAVs, communications realized with the proprietary protocol AirMax. The RSSI scan algorithm is a two-stage algorithm in which the first step finds a first orientation via a coarse-grained scanning. From this first orientation, the second step reduces the scan step as the search space. The algorithm is designed for a link with 2 UAVs that are able to schedule the start of the algorithm: as one UAV runs the algorithm, the second UAV waits for the execution completion before starting its own run of the algorithm. This synchronization is realized with handshaking signals (via the Xbee system). MATLAB simulations and real experiments are carried out on a scenario with 2 UAVs. The same authors study the same problem in [11], but, in this paper, the authors use a reinforcement learning approach to learn the communication channel model. The proposed solution is validated with 2 UAVs and with a focus on the reached angles and the learned antenna radiation pattern. These two studies are the closest to our work but they differ from ours on different aspects: we consider the WiFi communication protocol and more general scenarios with possibly more than 2 UAVs; the antenna orientation algorithm is local to each UAV without any synchronization between UAVs; we consider the possible use, by the WiFi interface, of a rate adaptation algorithm that may significantly impact the communication performance.

In some papers, the authors consider the possible mobility of the devices to improve the communication performance. This is for instance the case in [6, 12]. But in most cases, terrestrial robots are considered and the objective is to move some robots in some "good" locations in order to obtain more efficient communications. Very often, the antenna orientation and the WiFi rate adaptation are not taken into account as we do in our study.

6 CONCLUSION

The antenna radiation pattern is clearly an important factor when looking at the performances of a communication network. In this paper, we have underlined that, even without knowing the radiation pattern of its antennas, nor the positions of its peers, thanks to our antenna orientation algorithm, a node can improve its own network performances, as well as the ones of the whole network. However, this improvement depends on the underlying rate adaptation algorithms used by the WiFi interfaces, as well as the radiation patterns of the antennas. While this conclusion is limited to the studied simulation scenarios and the considered simulation environment, we believe that, given small modifications of the orientation algorithm, the conclusions would translate well in the real world.

REFERENCES

- [1] Mahmoud Badi, John Wensowitch, Dinesh Rajan, and Joseph Camp. 2019. Experimental Evaluation of Antenna Polarization and Elevation Effects on Drone Communications. In *Proceedings of the 22nd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. 211–220.
- [2] Sabur Baidya, Zoheb Shaikh, and Marco Levorato. 2018. FlyNetSim: An open source synchronized UAV network simulator based on ns-3 and ardupilot. In *Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. 37–45.
- [3] Constantine A Balanis. 2016. *Antenna theory: analysis and design*. John Wiley & sons.
- [4] Jienan Chen, Junfei Xie, Yixin Gu, Songwei Li, Shengli Fu, Yan Wan, and Kejie Lu. 2017. Long-range and broadband aerial communication using directional antennas (ACDA): Design and implementation. *IEEE Transactions on Vehicular Technology* 66, 12 (2017), 10793–10805.
- [5] F. Fietkau. [n.d.]. Minstrel HT: New rate control module for 802.11n. <https://lwn.net/Articles/376765/>.
- [6] Mahanth Gowda, Ashutosh Dhekne, and Romit Roy Choudhury. 2016. The case for robotic wireless networks. In *Proceedings of the 25th International Conference on World Wide Web*. 1317–1327.
- [7] Rémy Grünblatt, Isabelle Guérin-Lassous, and Olivier Simonin. 2019. Simulation and Performance Evaluation of the Intel Rate Adaptation Algorithm. In *Proceedings of the 22nd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. 27–34.
- [8] Yixin Gu, Mi Zhou, Shengli Fu, and Yan Wan. 2015. Airborne WiFi networks through directional antennae: An experimental study. In *2015 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 1314–1319.
- [9] Samira Hayat, Evşen Yanmaz, and Christian Bettstetter. 2015. Experimental analysis of multipoint-to-point UAV communications with IEEE 802.11 n and 802.11 ac. In *2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 1991–1996.
- [10] Rajendra K Jain, Dah-Ming W Chiu, William R Hawe, et al. 1984. A quantitative measure of fairness and discrimination. *Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA* (1984).
- [11] Songwei Li, Chenyuan He, Mushuang Liu, Yan Wan, Yixin Gu, Junfei Xie, Shengli Fu, and Kejie Lu. 2019. Design and implementation of aerial communication using directional antennas: learning control in unknown communication environments. *IET Control Theory & Applications* 13, 17 (2019), 2906–2916.
- [12] Karen Miranda, Nathalie Mitton, and Tahiry Razafindralambo. 2015. On the Impact of Routers' Controlled Mobility in Self-Deployable Networks.
- [13] Ubiquiti Networks. 2019. UAP Antenna Radiation Patterns. <http://web.archive.org/web/20200613210252/https://help.ui.com/hc/en-us/articles/115005212927-UniFi-UAP-Antenna-Radiation-Patterns>
- [14] Hang Qi, Zhiqun Hu, Xiangming Wen, and Zhaoming Lu. 2019. Rate Adaptation With Thompson Sampling in 802.11ac WLAN. *IEEE Communications Letters* 23 (2019), 1888–1892.
- [15] Russell Stuart, Norvig Peter, et al. 2003. Artificial intelligence: a modern approach.
- [16] Evşen Yanmaz, Samira Hayat, Jürgen Scherer, and Christian Bettstetter. 2014. Experimental performance analysis of two-hop aerial 802.11 networks. In *2014 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 3118–3123.
- [17] Evşen Yanmaz, Robert Kuschig, and Christian Bettstetter. 2013. Achieving air-ground communications in 802.11 networks with three-dimensional aerial mobility. In *2013 Proceedings IEEE INFOCOM*. IEEE, 120–124.
- [18] Nicola Roberto Zema, Angelo Trotta, Enrico Natalizio, Marco Di Felice, and Luciano Bononi. 2018. The CUSCUS simulator for distributed networked control systems: Architecture and use-cases. *Ad Hoc Networks* 68 (2018), 33–47.